MRUC 20
MRER 21
MRES 22

Malgré tout le soin apporté à l'élaboration de ce document, certaines imperfections peuvent subsister. La société BALOGH ne sera en aucun cas responsable de leurs conséquences directes ou indirectes non plus que de l'interprétation erronée des informations techniques.

Réf. : Préliminaire 96/21/02

TABLE DES MATIERES

1. SYSTEME MODUL R® 1
2. CARACTERISTIQUES FONCTIONNELLES 3
2.1. Communication série 3
3. PROGRAMMATION 10
3.1. PROGRAMME INIT 10
3.2. PROGRAMME AUTOMATE. 10
3.3. MODE MULTITACHE 13
3.4. MODE MONOTACHE 14
4. RESEAU CAN 15
4.1. UNITE CENTRALE - MRUC 20 15
4.2. MODULE - MRER 21 16
4.3. BLOC D'ENTREES/SORTIES - MRES 22 21

1. PRESENTATION DU SYSTEME MODUL R®

Composé d'une unité centrale (MRUC 20) et de périphériques, le système MODUL R® assure principalement les fonctions suivantes :

1.1. SAISIE D'INFORMATIONS

- Informations TOR ou analogiques,
- Identification (RFID, Code-barres etc...),
- Messages sur réseaux ou lignes "série"

1.2. TRAITEMENT OU PRETRAITEMENT

Traitement des informations (rôle de Filtre entre le terrain et les niveaux supérieurs) en vue d'obtenir une réactivité conforme aux temps de cycle et susceptible de les améliorer.

1.3. COMMUNICATION ET DIALOGUE "HOMME - MACHINE"

La carte MRUC 20 est équipée de 2 réseaux CAN et de 2 liaisons série".
Sur chaque réseau CAN il est possible de raccorder 8 modules MRER 21 ; soit 16 voies "série".

1.4. GESTION LOCALE D'AUTOMATISMES

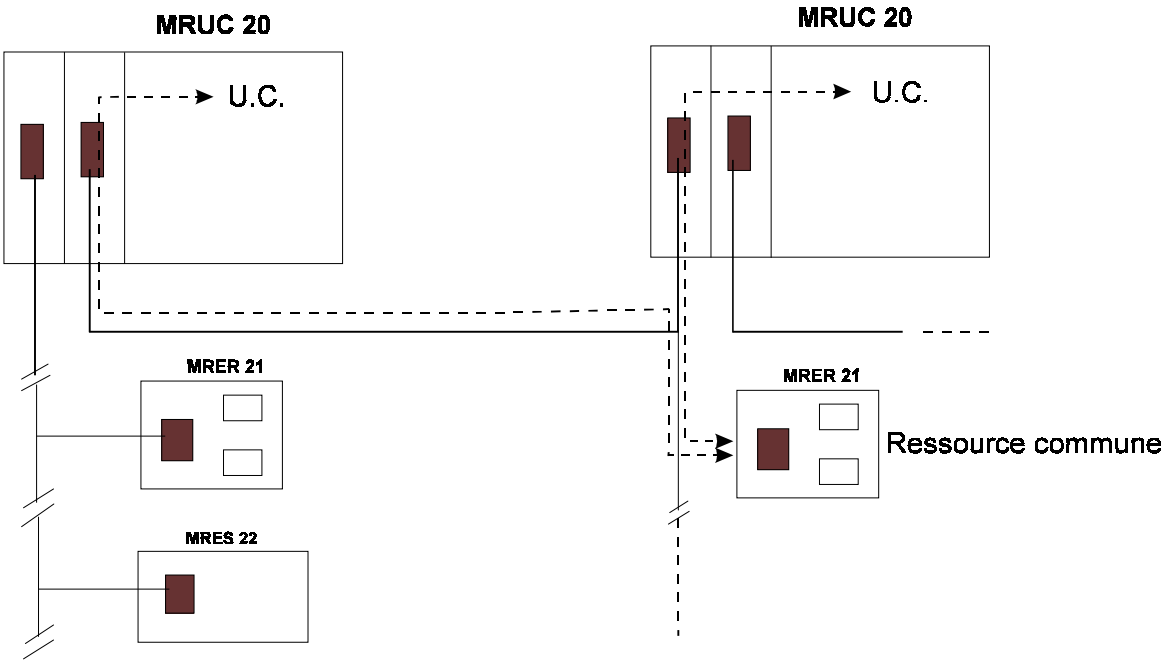
Chaque réseau CAN peut recevoir 16 modules de 16 entrées et 16 sorties T.O.R.

1.5. ARCHITECTURE ADAPTEE

L'utilisateur pourra associer les éléments du système MODUL. R selon l'architecture la mieux adaptée à ses besoins.

A cet effet nous présentons, ci-après, quelques exemples :

PARTAGE DE RESSOURCES



1.6. DOMAINES D'APPLICATION

- La productique ; en particulier l'équipement de postes ou de zones de travail (contrôle, mariage, encodage, retouches etc...) sur des lignes de fabrication,
- La logistique (magasins, pont-roulants, ferroviaire, aéroports),
- Le contrôle technique de bâtiments et contrôle d'accès

1.7. CONSTITUANTS DE BASE

- Unité centrale MRUC 20
  (carte encliquetable sur RAIL DIN avec borniers débrochables et 1 connecteur CANON 9 points
  dimension = largeur : 170 mm - hauteur : 122 mm- profondeur : 65 mm)
  2 réseaux CAN
  1 liaison RS 422 / 485 multiprotocole
  1 liaison RS 232 multiprotocole (le connecteur CANON)
  1 unité centrale programmable (RAM de 96 Ko)

- Module MRER 21
  (encliquetable sur RAIL DIN avec borniers débrochables.
  Dimensions = Largeur : 95 mm - hauteur : 95 mm - profondeur : 65 mm)
  2 voies "série" pour :
  Emetteur - récepteurs (lecture / écriture d'étiquettes RFID)
  Terminaux "écrans / claviers"
  Douchettes / imprimantes code-barres
  Capteurs "évolués"
  etc...

- Module MRES 22
  (encliquetable sur RAIL DIN avec borniers débrochables.
  Dimensions = largeur : 150 mm - hauteur : 96 mm - profondeur : 65 mm)
  - 16 entrées T.O.R. visualisées et protégées
  - 16 sorties T.O.R. visualisées et protégées

2. CARACTERISTIQUES FONCTIONNELLES

2.1. Communication série

La carte MRUC 20 est équipée de 2 liaisons série indépendantes dites UART0 et UART1.

2.1.1. Raccordement et type de liaison

2.1.1.1.UART 0:

La liaison est de type RS422/485. Elle pourra être utilisée en RS232 (restriction de tension = 0 à 12 Volts) sur la quasi totalité des équipements.

- **Utilisation en RS 422**

  Raccordements : bornier 4
  - borne 1 : Rx+
  - borne 2 : Rx-
  - borne 3 : Tx+
  - borne 4 : Tx-
  - borne 5 : 0 Volts UART 0
  - borne 6 : blindage

  RS422 3 ETATS (Multipoint)
  - Switch rouge = Point non apparent

  RS422 2 ETATS (point à point)
  - Switch rouge = Point apparent

- **Utilisation en RS 485 (Multipoint)**

  Raccordements : bornier 4
  - borne 1 (Rx+) strappée avec la borne 3 (Tx+)
  - borne 2 (Rx-) strappée avec la borne 4 (Tx-)
  - borne 5 : 0 Volts UART 0
  - borne 6 : blindage

  La liaison RS485 est une liaison 3 fils (câble blindé):
  - borne 1 ou 3 vers borne A de l'équipement utilisateur
  - borne 2 ou 4 vers borne B de l'équipement utilisateur
  - borne 5 vers 0 Volts de l'équipement utilisateur
  - borne 6 vers blindage du câble de liaison.

  La liaison RS485 est multipoint : Switch rouge en position point non apparent.

- **Utilisation en RS 232**

  Raccordements : bornier 4
  - borne 2 (Rx-) = Rx RS232
  - borne 4 (Tx-) = Tx RS232
  - borne 5 : 0 Volts de la liaison
  - borne 6 : blindage

  Remarques:
  - Les bornes 1 et 3 ne sont pas utilisées
  - La position du switch rouge n'a aucune importance.

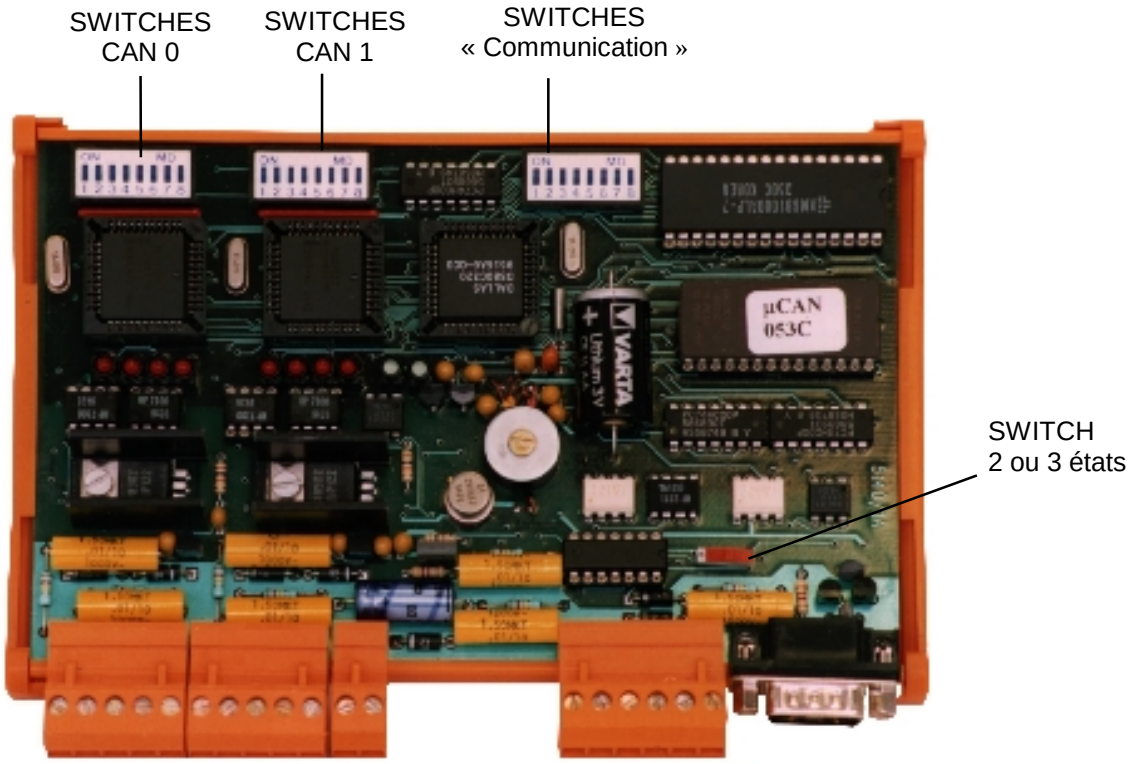
2.1.1.2.UART 1

Liaison exclusivement RS232. L'utilisation de cette voie sous une autre forme de liaison série nécessite un convertisseur (nous consulter).

Raccordement : connecteur CANON 9 points (mâle)
- broche 2 : Rx
- broche 3 : Tx

broche 5 : 0 Volts UART1 (ISOLE)

2.1.2. CONFIGURATION



2.1.2.1.Adresse de la carte MRUC20

Le numéro d'esclave de la carte est fixé par les switches « Communication »1 à 4.

SW1	SW2	SW3	SW4	NE = Numéro d'Esclave	
				Protocole BALOGH	Autres Protocoles
OFF	OFF	OFF	OFF	01H	10H
ON	OFF	OFF	OFF	03H	01H
OFF	ON	OFF	OFF	05H	02H
ON	ON	OFF	OFF	07H	03H
OFF	OFF	ON	OFF	09H	04H
ON	OFF	ON	OFF	0BH	05H
OFF	ON	ON	OFF	0DH	06H
ON	ON	ON	OFF	0FH	07H
OFF	OFF	OFF	ON	11H	08H
ON	OFF	OFF	ON	13H	09H
OFF	ON	OFF	ON	15H	0AH
ON	ON	OFF	ON	17H	0BH
OFF	OFF	ON	ON	19H	0CH
ON	OFF	ON	ON	1BH	0DH
OFF	ON	ON	ON	1DH	0EH
ON	ON	ON	ON	1FH	0FH

2.1.2.2.Parité, vitesse de transmission et choix du protocole

- **Protocole BALOGH ou JBUS (Switches COMMUNICATION).**

Il est utilisé pour la programmation, la configuration de la carte et le dépannage du programme. Ce protocole est validé par le switch 5 en position OFF pour l'UART 0 et par le switch 6 en position OFF pour l'UART 1. Dans ce cas les paramètres de communications sont fixés comme suit:

9600 Bds, parité IMPAIRE, 8 bits de données, 1 bit de stop.

- **Autres protocoles (mémorisés).**

Les adresses 5FEH (UART 0) et 5FFH (UART 1) contiennent une valeur qui défini le protocole utilisé et les paramètres de la liaison série. La carte prend en compte cette configuration pour l'UART 0 si le switch 5 est sur ON et si le switch 6 est sur ON pour l'UART 1.
La carte ne prend en compte les modifications qu'a la mise sous tension.

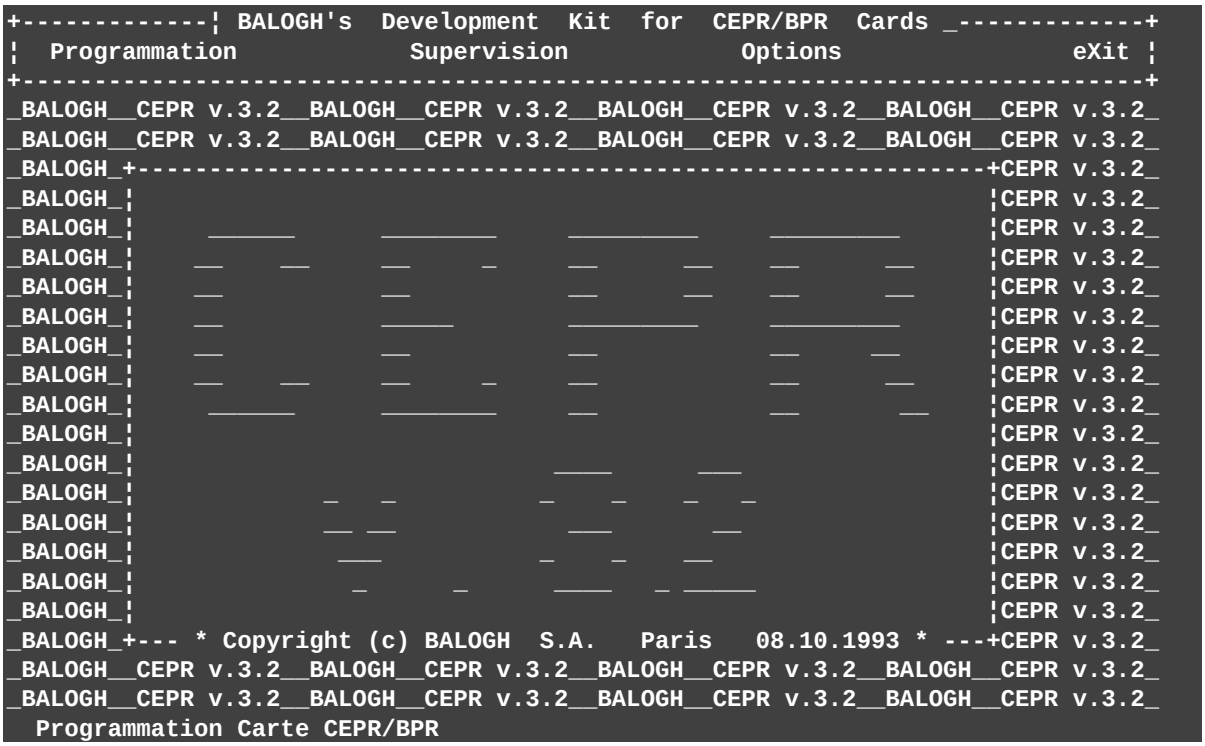
Description d'un octet de configuration (adresse 5FEH ou 5FFH)

b7	b6	b5	b4	vitesse		b3	b2	b1	b0	Protocole
parité				Bds						
0 = ODD 1 = EVEN	0	0	0	19200		0	0	0	0	BALOGH/JBUS®
	0	0	1	9600		0	0	0	1	UNITELWAY®
	0	1	0	4800		0	0	1	0	3964R®
	0	1	1	2400		0	1	0	0	DF1 Half duplex®
	1	0	0	1200		0	1	0	1	DF1 Full duplex®
	1	0	1	600		0	1	1	0	Pour extension
	1	1	0	300		0	1	1	1	pour extension
	1	1	1	150		1	0	0	0	Protocole libre données sur 8 bits
						1	0	0	1	Protocole libre données sur 7 bits

2.1.2.3.Test de dialogue avec la carte MRUC20

Raccorder un PC sur la voie UART 1 (connecteur CANON) et utiliser le logiciel CEPR

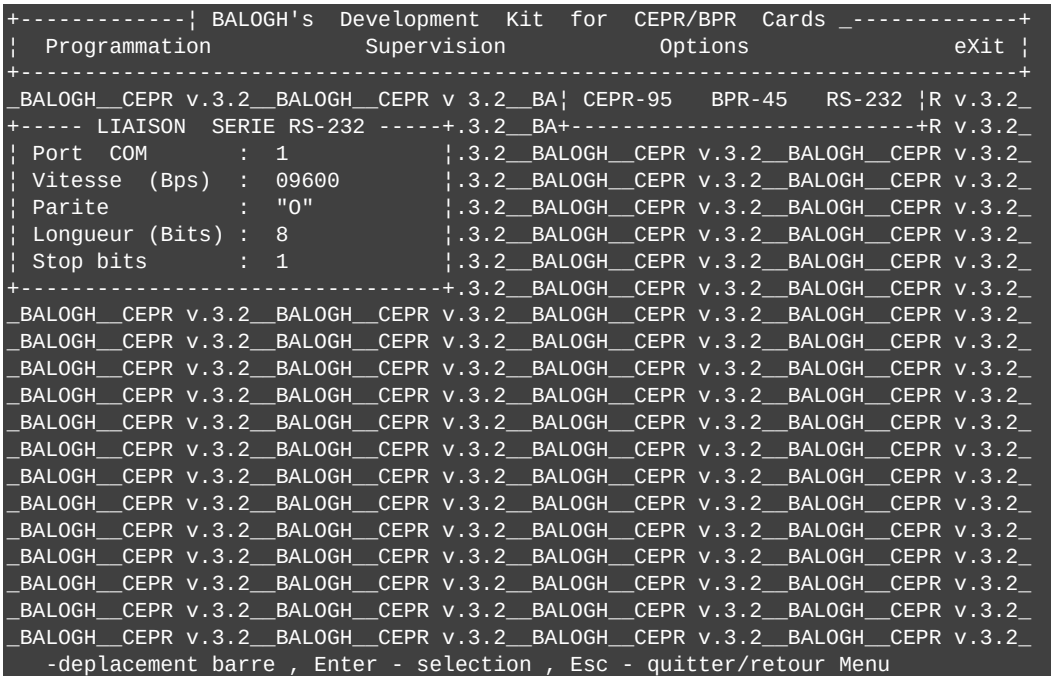
Accueil du logiciel CEPR.



Configuration de la liaison RS232 du PC

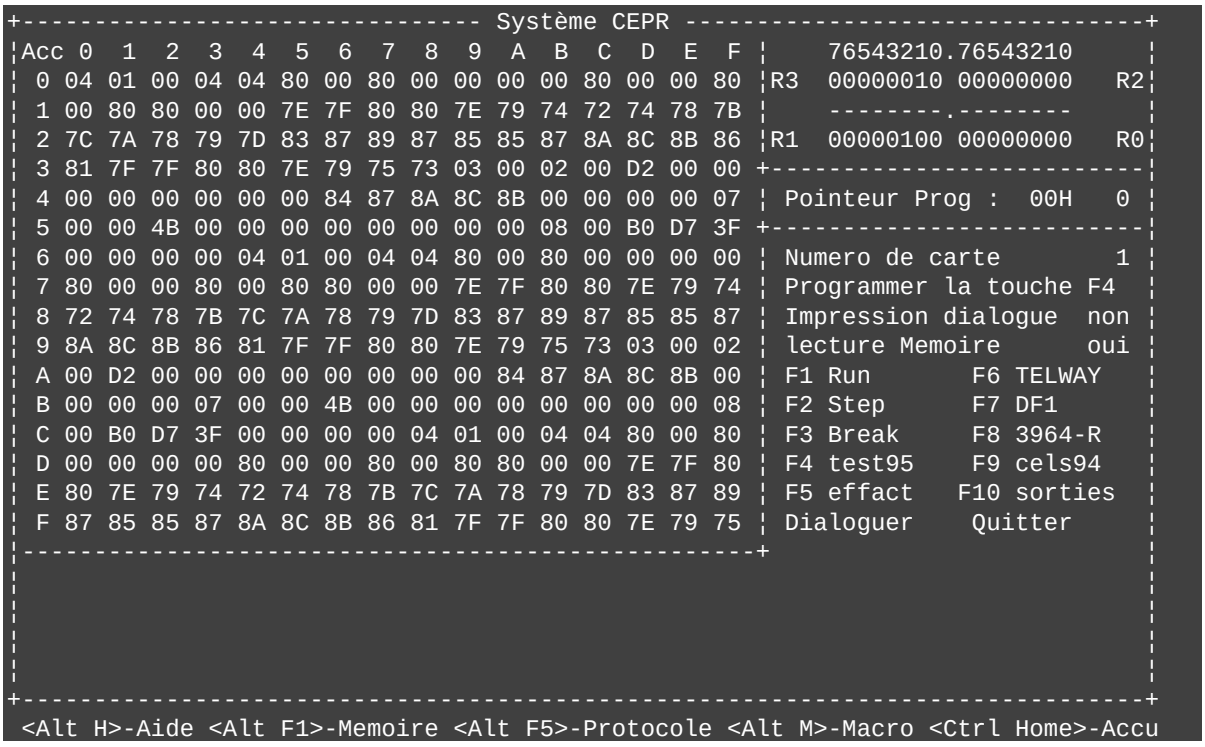
Dans le menu « option » choisir RS232, une fenêtre contenant les paramètres est affichée. Choisir la configuration pour le protocole BALOGH : N° du port série du PC (1 = COM1), vitesse = 9600, Parité ODD = « O », 8 bits de données, 1 stop.

Configuration du port de communication du PC.



Une fois la liaison correctement configurée, choisir le menu « Programmation » puis le menu « Dialoguer ».

Ecran en mode dialogue.



Pour vérifier que la liaison est correcte, lire par exemple les adresses 5FEH et 5FFH contenant respectivement les paramètres de la liaison série UART 0 et UART 1.
Pour taper la ligne de commande, passer en mode dialogue (taper sur la touche « d » ou choisir le menu « Dialoguer »).
Commande pour la lecture de 2 octets à partir d'une adresse
Mnémonique : LLD,2,5FEH.
Trame :INS,2CH,2,5FH,EH.

Réponse de la carte :
01,2C,FF,0A,02,05,FE,pF,pf,04.

pF = <5FEH> = paramètres mémorisés pour le protocole de l'UART 0.
pf = <5FFH> = paramètres mémorisés pour le protocole de l'UART 1

Si la carte ne répond pas :
Vérifier que la carte est bien alimentée,
Vérifier la liaison (câble croisé ou non),
Vérifier les paramètres de transmission,
Vérifier que le N° de carte indiqué à l'écran est identique à celui codé sur la carte,
Vérifier que le switch 6 est en position OFF (UART 1 sous protocole BALOGH)
Vérifier la ligne de commande,

Si la carte répond, il est maintenant possible de mémoriser une configuration spécifique pour la ou les voies séries.

Exemple : mémorisation du protocole libre sur l'UART 0 avec les paramètres de communication suivants : parité paire, 9600 Bds. La valeur à écrire à l'adresse 5FEH est 98H. (Cf. 1.22 Parité, vitesse de transmission et choix du protocole)

Commande :
Mnémonique : LMOV,1,5FEH,98H.
Trame : INS,72H,1,5FH,EH,98H.

Réponse :
01,72,FF,09,01,05,FE,98,04.

La carte doit répondre, si elle ne répond pas voir procédure précédente.

2.1.3. EMISSION/RECEPTION DE CARACTERES (PROTOCOLE LIBRE)

Le protocole libre étant mémorisé aux adresses 5FEH pour l'UART 0 et 5FFH pour l'UART 1. Basculer le switch 5 sur ON pour valider le protocole de l'UART 0 et/ou le switch 6 sur ON pour valider le protocole de l'UART 1. Mettre la carte Hors/Sous tension pour que les modifications soient prises en compte.

2.1.3.1.EMISSION

Les liaisons séries sont vues par la carte comme des esclaves :
La liaison série UART 0 = esclave 20H
La liaison série UART 1 = esclave 21H

L'émission revient à écrire sur l'esclave avec une instruction:
écriture « adressage direct » IW (=INS,63H,...)
écriture « adressage indirect » IW.I (=INS,67H,...)

Exemple :
Emission sur UART 0, 9600 Bds, parité paire d'un message dont le premier octet est stocké à l'adresse 4000H. Le status sera stocké à l'adresse 820H.

Conditions initiales :
l'adresse 5FEH doit contenir 98H,
Le switch est sur ON

Commande:
Mnémonique : IW,20H,128,xx,4000H,820H.
Trame : INS,63H,20H,0,80H,x,x,40H,0,8,20H.
x étant une valeur quelconque.

L'émission est en cours tant que <820H> = 0xxxxxxx.
Emission terminée sans défaut si <820H> = 80H
Emission abandonnée sur défaut si <820H> = 9FH (défaut = autre émission déjà en cours)

2.1.3.2.RECEPTION

Lorsqu'un protocole libre est sélectionné, les données reçues sont stockées automatiquement dans un buffer circulaire:
adresses 0000H à 00FFH pour l'UART 0
adresses 0200H à 02FFH pour l'UART 1.
Ces buffers sont pointés respectivement par les valeurs contenues aux adresses 0513H et 0515H qui indiquent la position de stockage de la prochaine donnée à stocker.

La longueur du message doit être contrôlée par l'utilisateur : position à tn - position (tn-1).
La valeur du pointeur est modulo FFH soit lorsque le pointeur prend la valeur FFH, elle est remplacée par 00H (retour au début du buffer et écrasement des anciennes valeurs).

Exemple : Réception de caractères émis par un clavier.
La voie UART 0 est configurée en protocole libre et le switch 5 sur ON.
A la mise sous tension ou par programme, le pointeur du buffer de la voie UART 0 est effacé (<0513H> = 00).
5 caractères sont émis par le clavier:

Les caractères sont rangés aux adresses 0000H à 0004H
Le pointeur passe à 05H. (0513H = 05H)

6 caractères sont de nouveau émis par le clavier :

Les caractères sont rangés aux adresses 0005H à 000AH
Le pointeur passe à 0BH (0513H = 0BH)

Etc ...

3. PROGRAMMATION

La programmation de la carte MRUC20 est effectuée au travers du logiciel CEPR pour PC. Création du programme, compilation, transfert vers la mémoire RAM de la carte (96 Koctets dont 32 accessibles en supervision) et débuggage de l'application.

Le programme sera composé d'une partie appelée « PROGRAMME INIT » (254 lignes Max) et d'une partie principale (4 bancs de 254 lignes en mode MONOTACHE ou 4 programmes de 254 lignes en mode MULTITACHE).

A la mise sous tension, l'UC exécute le programme INIT (1 seule fois), puis la partie principale.

3.1. PROGRAMME INIT

Il contient de 1 à 254 lignes de programme. Les lignes de 0 à ORG-1 sont considérées comme du commentaire (rien ne sera exécuté dans cette zone).
A partir de la ligne ORG, les instructions doivent être exécutable et respecter la syntaxe BALOGH du logiciel CEPR.

Le programme INIT doit contenir la définition de chacun des modules présents sur chacune des voies CAN (MRUC20, MRER 21 ou MRES 22).
La sélection d'un banc de programme ou la déclaration du mode MULTITACHE met fin au programme INIT et démarre l'application programmée.

Transfert du programme INIT dans la mémoire de la carte:

Sélectionner le mode dialogue du logiciel CEPR
Taper sur la touche de fonction « BREAK » (arrêt du programme utilisateur de la carte)
Taper la commande de sélection du programme INIT:
INS,7FH,3FH.
Quitter le mode Dialogue en pressant la touche « Echap »
Charger le programme compilé et le télécharger (commande Envoyer fichier)

3.2. PROGRAMME AUTOMATE.

Le programme automate est défini par les lignes 0 à ORG-1. Le programme automate est exécuté toutes les 20 ms en interruption du programme principal.

Quelques remarques importantes :
L'ordre d'écriture des lignes fixe la priorité pour les interruptions en cours.

Quand une interruption est en cours d'exécution (durée > 20ms), les interruptions suivantes ne tiennent pas compte des lignes qui suivent celle en cours; et cela tant que l'interruption en cours n'est pas terminée (par l'instruction CI).

Le module MRUC20 peut gérer 8 niveaux d'interruptions et/ou de sous programmes.

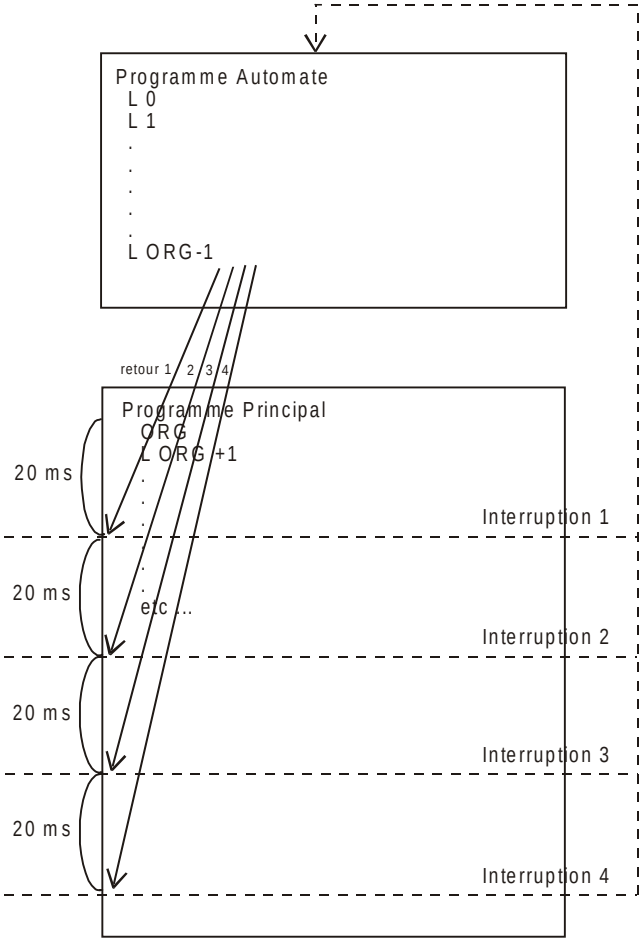
Une interruption ou un sous programme doit toujours être terminé. Si pour une raison quelconque, l'utilisateur souhaite réaliser un reset par programme, il devra utiliser l'instruction CSP et sauter à la ligne de début.

Le temps passé dans le programme automate, donc le nombre de ligne programme, sera le plus court possible, étant donné que le déroulement du programme principal est alors suspendu.

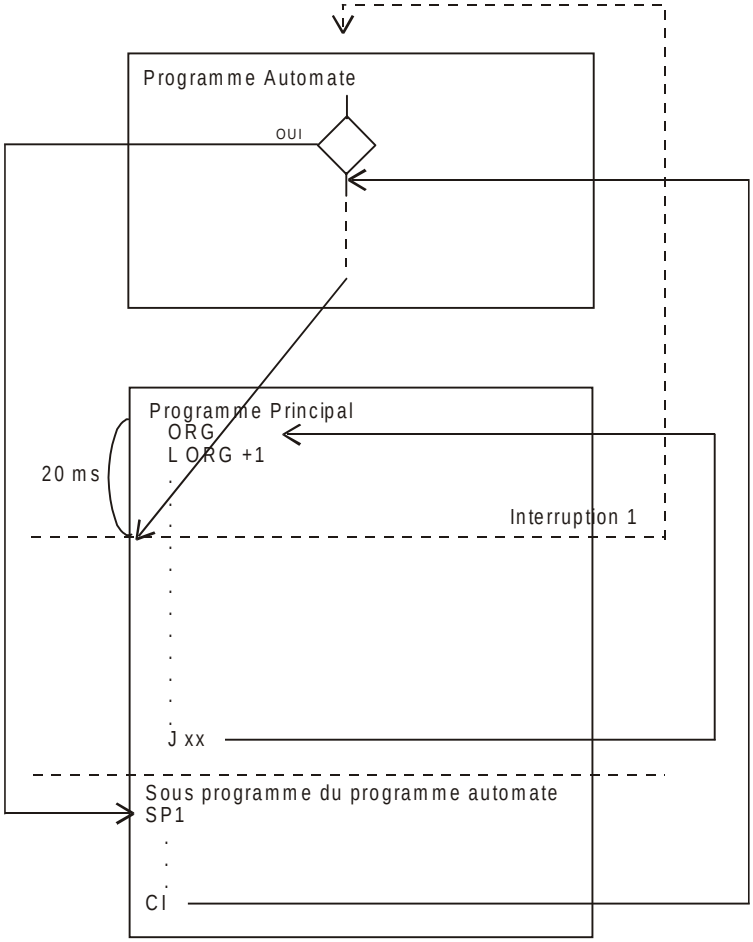
Lors d'une demande en supervision sur une ligne série, le programme ou les interruptions en cours sont interrompus pour exécution immédiate de la consigne demandée.

L'utilisateur prendra soin de boucler son programme principal afin de ne pas entrer intempestivement dans une interruption.

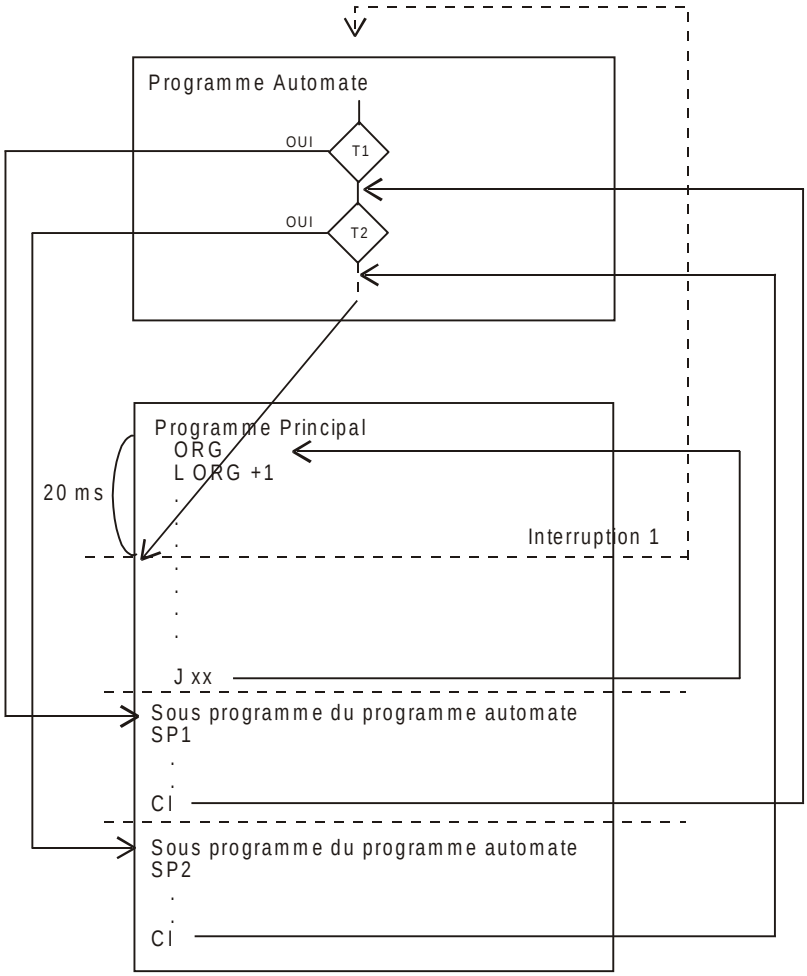
Ex1 : Programme automate sans sous programme et dont le temps d'exécution est < 20ms.



Ex 2 : Programme automate appelant un sous programme.



Ex 3 : Programme automate avec gestion de priorités



3.3. MODE MULTITACHE

Chaque programme (0,1,2 ou 3) dispose d'une zone mémoire spécifique de 256 octets dite ACCUMULATEUR et de son propre programme AUTOMATE.

ACCU programme 0	=>	0800H-08FFH
ACCU programme 1	=>	0900H-09FFH
ACCU programme 2	=>	0A00H-0AFFH
ACCU programme 3	=>	0B00H-0BFFH

Les instructions de changement de banc ou d'ACCU sont ignorées.
Les échanges éventuels d'informations entre programme sont réalisés à l'aide des instructions de manipulation de registres (SET.., IN.., OUT..) ou les instructions 16 bits (L....).

Attention : La mémoire RAM (96Ko), les registres d'entrées/sorties, les tempos et de façon générale toutes les ressources de l'UC doivent être partagées.

3.3.1. Organisation des programmes.

Si le programme INIT contient l'instruction de sélection du mode MULTITACHE (INS,7FH,3nH.) l'UC exécute n+1 programme en parallèle soit :

SEL,30H. (n=0)	=> exécution du programme 0.
SEL,31H. (n=1)	=> exécution des programmes 0 et 1.
SEL,32H. (n=2)	=> exécution des programmes 0,1 et 2
SEL,33H. (n=3)	=> exécution des programmes 0,1,2 et 3.

Chaque programme contient de 1 à 254 lignes. Les lignes 0 à ORG-1 sont définies comme un programme AUTOMATE qui est exécuté toutes les 20ms en interruption du programme PRINCIPAL qui débute à la ligne ORG.
Si la ligne ORG est la ligne 0, il n'y a pas de programme AUTOMATE.
Les appels aux sous programmes (instruction CALL) ouvrent des interruptions et doivent impérativement se terminer par l'instruction CI (que l'appel provienne du programme PRINCIPAL ou du programme AUTOMATE).

3.3.2. Ecriture et téléchargement du programme.

Chaque programme est écrit, compilé et téléchargé individuellement.

3.3.2.1.Chargement du programme INIT avec le logiciel CEPR:

Mode Dialogue
- BREAK (arrêt de la carte)
- SEL,3FH. (sélection du programme INIT)

Q (quitter)
Mode Transmission
- Envoyer fichier « Nom du programme INIT compilé »

3.3.2.2.Chargement des programmes utilisateur

Mode dialogue
- BREAK (arrêt de la carte)
- SEL,n. (sélection du programme à télécharger 0,1,2 ou 3)

Q (quitter)
Mode transmission
- Envoyer fichier « Nom du programme compilé correspondant »

Remarque : La présence de programme vide influe sur la vitesse exécution des autres programmes.

3.4. MODE MONOTACHE

Lorsque le programme comprend plus de 254 lignes de programme, il peut être réparti sur 3 bancs supplémentaires de 254 lignes.

Ce mode de fonctionnement est déclenché par un programme INIT vide (le programme débute alors à la ligne ORG du banc 0) ou par l'instruction SEL,BANC n. (le programme débute alors à la ligne ORG du banc n).

Le programme AUTOMATE de l'ensemble de l'application est stocké dans le banc de démarrage et utilise automatiquement l'ACCU 0 (0800H-08FFH). **Les lignes 0 à ORG-1 des autres bancs sont ignorées.**

Le banc de démarrage contient la boucle de fond du programme alors que les autres bancs contiennent des sous programmes ou des blocs de code qui sont accessibles par des instructions de sauts (J ou CALL). L'utilisation de l'instruction CALL implique un retour par l'instruction CI (dans ce cas le bloc d'appel est mémorisé et ligne de retour = ligne d'appel + 1)

3.4.1. Téléchargement du programme.

Le programme est chargé dans la carte banc par banc. La sélection du banc est réalisée par l'instruction SEL,x.:

Affectation des bits de x pour l'instruction SEL								**Commentaires**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
0	0	0	0	Num ACCU		Num Banc		Démarrage du programme sur le banc Num Banc avec l'ACCU Num ACCU
0	0	0	1	Num ACCU		x	x	Chargement de l'ACCU Num accu
0	0	1	0	x	x	Num Banc		Démarrage du programme sur le banc Num banc avec l'ACCU courant
0	0	1	1	0	0	Nb de programme		Démarrage du mode multitâche sur Nb de programme
0	0	1	1	1	1	1	1	Sélection du banc INIT. Valable seulement en supervision

x = indifférent

3.4.1.1. Téléchargement du programme INIT

Mode Dialogue
BREAK.
SEL,3FH. (sélection du Banc INIT)
Q (quitter)
Mode Transmission
Envoyer fichier « Nom du programme INIT compilé »

3.4.1.2. Téléchargement du programme sur chacun des bancs

Mode Dialogue
BREAK.
SEL,n. (n = 0à 3, sélection du Banc)
Q (quitter)
Mode Transmission
Envoyer fichier « Nom du programme compilé relatif au banc sélectionné »

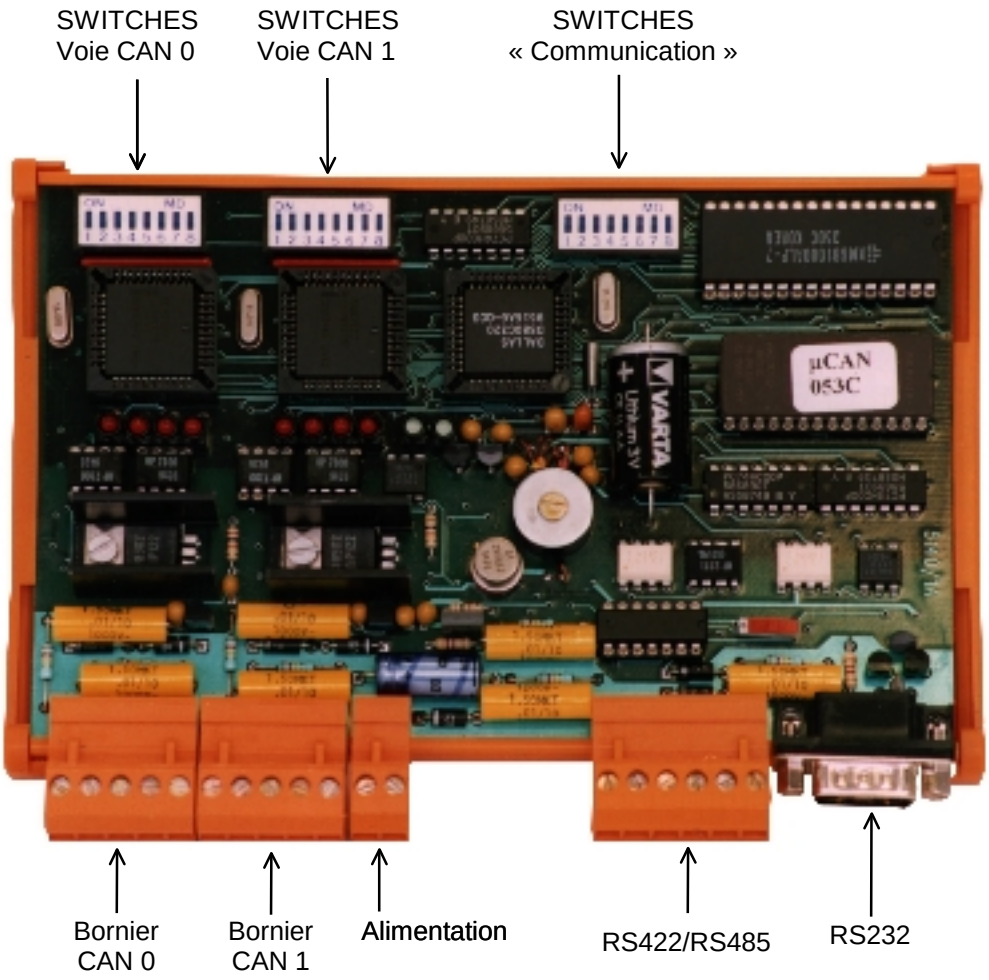
4. RESEAU CAN

La carte MRUC20 est équipée de 2 réseaux CAN dit CAN0 et CAN1.
Sur chacun des 2 réseaux on peut connecter :

8 modules « double voies » E/R ou UART réf : MRER 21

16 modules d'entrées/sorties tout ou rien (16 entrées, 16 sorties par module). Réf : MRES 22.

4.1. UNITE CENTRALE - MRUC 20



Affectation des LEDS.

1	2	3	4

5	6	7	8

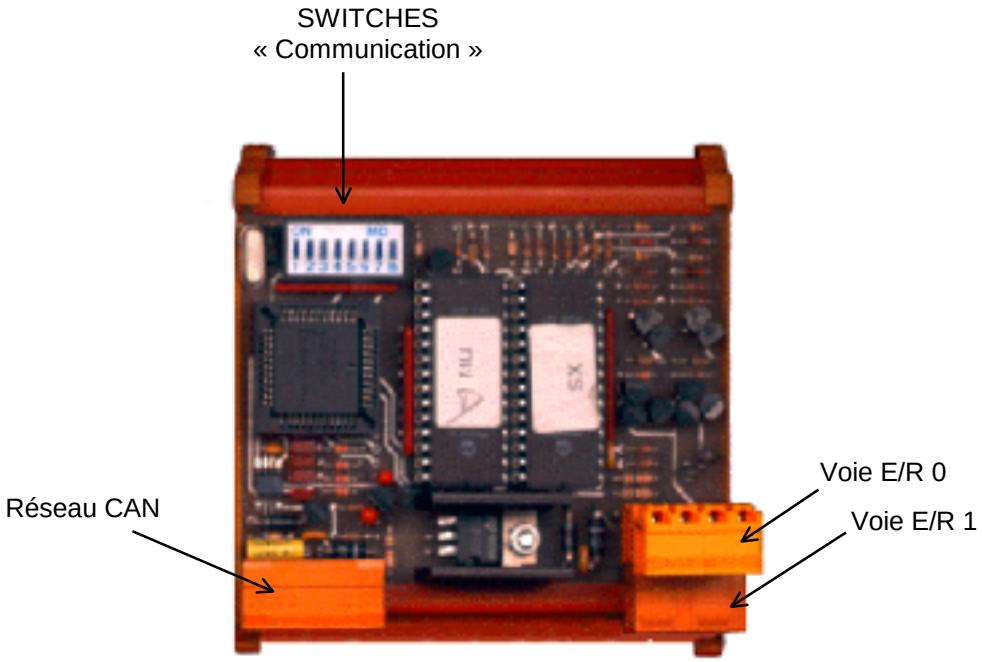
9	10

LED 1 à 4 : Réseau CAN 0
LED 5 à 8 : Réseau CAN 1
LED 9 :
LED 10 :

IMPORTANT :

- Si l'un des 2 réseaux n'est pas utilisé, le strapper fil à fil sur l'autre.
- Les 2 réseaux CAN doivent être alimentés.

4.2. MODULE - MRER 21



Pour les raccordements Cf fiche technique MRER 21

4.2.1. CONFIGURATION

Pour être reconnu sur le réseau chaque module est associé à **un numéro d'esclave différent**. Le numéro d'esclave du module est codé par les switches 1 à 4 (code binaire de 0 à 7).

Le MRER21 possède 2 processeurs (1 pour chaque voie) qui disposent chacun de 2 buffers de communication pour servir 2 clients (2 maîtres MRUC20). Une adresse physique est assignée à chacune des voies du module. Une adresse physique est encodée comme suit sur 1 octet:

bit 7	toujours = 1 (bit de validation du module)	
bit 6	Surveillance	= 1 pour mise à jour permanente du status
		= 0 pour status non mis à jour
bit 5	Processeur	= 1 pour processeur esclave
		= 0 pour processeur maître
bit 4	Buffer	= 0 pour buffer 0 (client 0). **Toujours 0 pour une voie UART**.
		= 1 pour buffer 1 (client 1)
bit 3,2,1,0	N° esclave	= recopie des switches 1 à 4 du module MRER21.

4.2.2. DECLARATION DES MODULES MRER21

Les modules sont déclarés et utilisés sur le réseau sous forme de voies logiques. Les adresses physiques sont associées aux voie logiques par les instructions :

INS,64H,AdPh0, AdPh 1,...., AdPh 15. pour la voie CAN0
INS,66H, AdPh 0, AdPh 1,...., AdPh 15. pour la voie CAN1

Les valeurs AdPh 0 à AdPh 15 sont les adresses physiques des modules (cf 3.21), les numéros de voies logiques sont attribuées dans l'ordre de saisie dans la ligne d'instruction.
Exemple:

Un réseau est composé de 2 modules MRER21 sur la voie CAN0. Chacun des modules comprend une voie OMX et une voie UART.
Le premier module est l'esclave 3 (switches 1 à 4), le second est l'esclave 5 (switches 1 à 4).

Adresses physiques:

Esclave 3:
Voie OMX en mode surveillance, processeur maître, utilisation du buffer 0
=> adresse physique C3H.
Voie UART en mode surveillance, processeur esclave, utilise obligatoirement le buffer 0
=> adresse physique A3H.
Esclave 5:
Voie OMX en mode surveillance, processeur esclave, utilisation du buffer 0.
=> adresse physique E5H.
Voie UART non surveillée, processeur maître, utilise obligatoirement le buffer 0.
=> adresse physique 85H.

Déclaration des modules

Ex 1: déclaration de module sur CAN 0
INS,64H,C3H,A3H,E5H,85H.

Voie OMX du module MRER 3 assignée à la voie logique 0.
Voie UART du module MRER 3 assignée à la voie logique 1
Voie OMX du module MRER 5 assignée à la voie logique 2
Voie UART du module MRER 5 assignée à la voie logique 3

Ex 2: déclaration de module sur CAN 0
INS,64H,C3H,E5H,A3H,85H.

Voie OMX du module MRER 3 assignée à la voie logique 0.
Voie OMX du module MRER 5 assignée à la voie logique 1
Voie UART du module MRER 3 assignée à la voie logique 2
Voie UART du module MRER 5 assignée à la voie logique 3

Ex 3: déclaration de module sur CAN 1
INS,66H,C3H,E5H,A3H,85H.

Voie OMX du module MRER 3 assignée à la voie logique 0.
Voie OMX du module MRER 5 assignée à la voie logique 1
Voie UART du module MRER 3 assignée à la voie logique 2
Voie UART du module MRER 5 assignée à la voie logique 3

La voie CAN 0 dispose de 16 voies logiques : 0 à FH.
La voie CAN 1 dispose de 16 voies logiques : 10H à 1FH.

Les voies logiques 0,1,2,3 pour CAN 0 et 10H,11H,12H,13H pour CAN 1 sont privilégiées. Dans le cas ou des voies demandent beaucoup de ressources, il est préférable de leur associer une de ces voies logiques.

Remarque: Pour accéder à une voie physique, les instructions utilisent le N° de la voie logique.

4.2.3. ZONE MEMOIRE RESERVEE POUR LES MODULES MRER21

Chacune des voies logiques déclarées sur le réseau dispose d'une zone mémoire de 16 octets contenant des informations relative à la voie:

Voie CAN 0	Voie CAN 1	Désignation
600H	700H	Status de la voie logique 0
601H	701H	Pointeur sur buffer de réception (si voie UART)
610H	710H	Status de la voie logique 1
611H	711H	Pointeur sur buffer de réception (si voie UART)
620H	720H	Status de la voie logique 2
621H	721H	Pointeur sur buffer de réception (si voie UART)
630H	730H	Status de la voie logique 3
631H	731H	Pointeur sur buffer de réception (si voie UART)
640H	740H	Status de la voie logique 4
641H	741H	Pointeur sur buffer de réception (si voie UART)
650H	750H	Status de la voie logique 5
651H	751H	Pointeur sur buffer de réception (si voie UART)
660H	760H	Status de la voie logique 6
661H	761H	Pointeur sur buffer de réception (si voie UART)
670H	770H	Status de la voie logique 7
671H	771H	Pointeur sur buffer de réception (si voie UART)
680H	780H	Status de la voie logique 8
681H	781H	Pointeur sur buffer de réception (si voie UART)
690H	790H	Status de la voie logique 9
691H	791H	Pointeur sur buffer de réception (si voie UART)
6A0H	7A0H	Status de la voie logique 10
6A1H	7A1H	Pointeur sur buffer de réception (si voie UART)
6B0H	7B0H	Status de la voie logique 11
6B1H	7B1H	Pointeur sur buffer de réception (si voie UART)
6C0H	7C0H	Status de la voie logique 12
6C1H	7C1H	Pointeur sur buffer de réception (si voie UART)
6D0H	7D0H	Status de la voie logique 13
6D1H	7D1H	Pointeur sur buffer de réception (si voie UART)
6E0H	7E0H	Status de la voie logique 14
6E1H	7E1H	Pointeur sur buffer de réception (si voie UART)
6F0H	7F0H	Status de la voie logique 15
6F1H	7F1H	Pointeur sur buffer de réception (si voie UART)

Important :

Le status du module n'est pas celui d'une instruction de lecture ou d'écriture mais l'état du module hors contrat. Les instructions ont leur propre status indiquant l'exécution avec ou sans défaut. de l'opération en cours.

4.2.4. PARTICULARITES DES MODULES UART.

Si la voie série n'est pas configurée (status = 80H), les messages reçus ne sont pas transmis à l'MRUC 20.
Si une émission est demandée, les paramètres par défaut sont : 9600 Bds, 8 bits, parité impaire (ODD), 1 stop.
La modification des paramètres émission et la réception éventuelle de message est obtenue après configuration.

Configuration de la voie UART:
L'instruction « IR » permet de configurer la ligne série en émission.
L'instruction « IRA » permet de configurer la ligne série en émission et en réception.
Le poids faible du paramètre Longueur, fixe le time out de la réception (unité = 0.1ms)
Le poids fort du paramètre Adresse, fixe les paramètres de réception (voir SM ci après)
Le poids faible du paramètre Adresse, fixe les paramètres émission (voir SM ci-après)

Octet de configuration (SM)

bit 7	type de parité	= 0 paire = 1 impaire
bit 6	parité	= 0 avec = 1 sans
bit 5	Nb de stop	= 0 1 stop = 1 2 stop
bit 4	format	= 0 8 bits = 1 7 bits
bit 3	non utilisé	= 0

vitesse	**9600**	**4800**	**2400**	**1200**	**600**	**300**	**150**	**75**
bit 2	0	0	0	0	1	1	1	1
bit 1	0	0	1	1	0	0	1	1
bit 0	0	1	0	1	0	1	0	1

Après une instruction IRA, le status de la voie et le pointeur sur le buffer de réception sont mis à jour par l'unité centrale (surveillance automatique).
Exemple : Configuration d'une voie UART (sur la voie logique 1).
Les paramètres de la liaison séries sont les suivants :
Time out 10ms (temps inter caractère considéré comme une fin de bloc)
Réception 7 bits, 1stop, parité paire, 4800 Bds.
Emission 8 bits, 1 stop, parité impaire, 2400 Bds.

Commande à envoyer (le status du contrat sera placé à l'adresse 880H).
Mnémonique IRA,1,100,1182H,X,X,880H.
Trame INS,63H,1,0,100,11H,82H,X,X,8,80H.
(X quelconque mais nécessaire)

Après exécution:
<880H> = E0H
<610H> = 60H
<611H> = 00H

- **Emission sur la voie UART:**
  Les données reçues sont rangée dans des buffers spécialisés (256 octets max.) de l'unité centrale:
  Le status du module et le pointeur du buffer sont mis à jour par l'unité centrale (surveillance automatique)

Voie CAN 0	Voie CAN 1	Désignation
1000H	2000H	Buffer de réception de l'UART logique 0
1100H	2100H	Buffer de réception de l'UART logique 1
1200H	2200H	Buffer de réception de l'UART logique 2
1300H	2300H	Buffer de réception de l'UART logique 3
1400H	2400H	Buffer de réception de l'UART logique 4
1500H	2500H	Buffer de réception de l'UART logique 5
1600H	2600H	Buffer de réception de l'UART logique 6
1700H	2700H	Buffer de réception de l'UART logique 7
1800H	2800H	Buffer de réception de l'UART logique 8
1900H	2900H	Buffer de réception de l'UART logique 9
1A00H	2A00H	Buffer de réception de l'UART logique 10
1B00H	2B00H	Buffer de réception de l'UART logique 11
1C00H	2C00H	Buffer de réception de l'UART logique 12
1D00H	2D00H	Buffer de réception de l'UART logique 13
1E00H	2E00H	Buffer de réception de l'UART logique 14
1F00H	2F00H	Buffer de réception de l'UART logique 15

Exemple: réception de 20 caractères sur la voie logique 1

En cours de réception :

<610H>	=	0110 0XXX	(b3 = 0 => réception en cours)
<611H>	<	14H	(pointeur sur buffer de réception)

Réception terminée :

<610H>	=	0110 1XXX	(b3=1 => time out)
<611H>	=	14H	(les 20 caractères sont stockés)

Les caractères sont rangés dans la plage 1100H à 1113H.

Exemple: Emission d'un message de 50 caractères stocké en mémoire à partir de l'adresse 3000H. Le status du contrat sera stocké à l'adresse 880H.

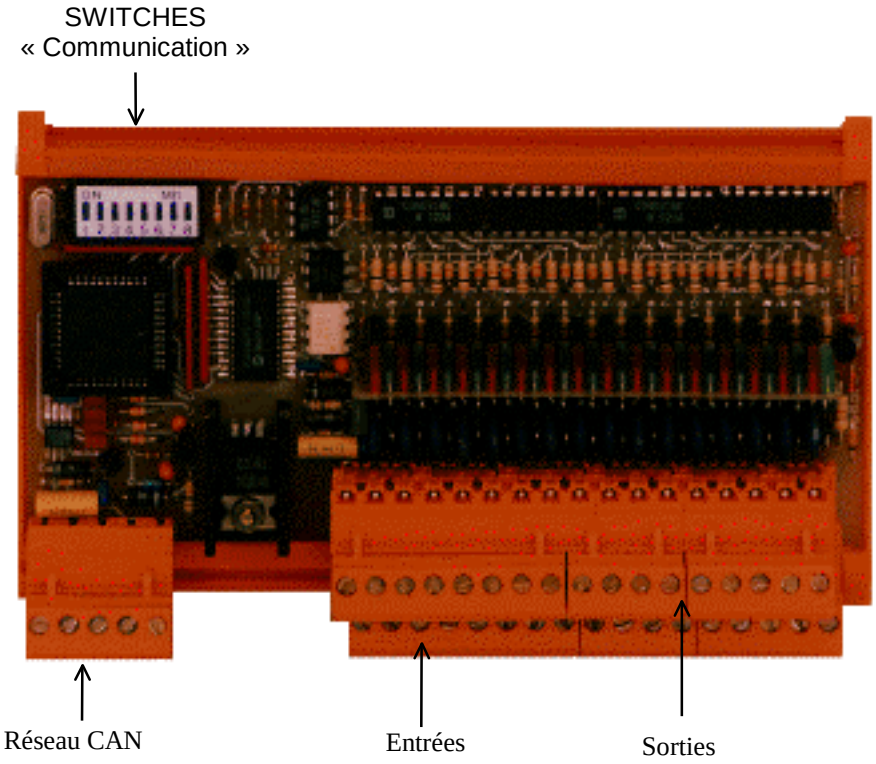
IW,1,50,X,3000H,880H.

Emission en cours : Le bit 7 du status = 0
Emission terminée : Le bit 7 du status = 1

Si b4 = 1 : la voie logique était déjà utilisée. Recommencer l'émission dès que la voie est libre.

4.3. BLOC D'ENTREES/SORTIES - MRES 22

Le module MRES 22, connecté au réseau CAN, permet de piloter 16 entrées/16 sorties. 16 modules MRES 22 peuvent être connectés sur un réseau CAN.



Pour les raccordements Cf. fiche technique MRER 21.

4.3.1. CONFIGURATION et DECLARATION

Le numéro d'esclave sur le réseau CAN est fixé par codage binaire des switches « Communication » 1 à 4 (0 à FH).

- Chaque bloc doit avoir son numéro propre.
- Chaque bloc possède 4 buffers de communication pour servir 4 clients (MRUC 20)

Une **adresse physique** définie comme suit, est affectée à chaque blocs MRES 22:

Bit 7	Toujours à 1 (bit de validation du module)
Bit 6	Toujours à 0
Bits 5,4	Codage binaire du numéro de buffer utilisé par l'Unité Centrale.
Bits 3 à 0	Recopie du numéro d'esclave du module MRES 22.

Chaque adresse physique ainsi définie doit être affectée à une adresse logique. Chaque adresse logique correspond à une adresse mémoire spécifique contenant l'adresse physique.

Réseau CAN 0		Réseau CAN 1		Désignation
Adresse logique	adresse de déclaration	Adresse logique	adresse de déclaration	
0H	0C0FH	10H	0D0FH	Adresse physique d'un module
1H	0C1FH	11H	0D1FH	Adresse physique d'un module
2H	0C2FH	12H	0D2FH	Adresse physique d'un module
3H	0C3FH	13H	0D3FH	Adresse physique d'un module
4H	0C4FH	14H	0D4FH	Adresse physique d'un module
5H	0C5FH	15H	0D5FH	Adresse physique d'un module
6H	0C6FH	16H	0D6FH	Adresse physique d'un module
7H	0C7FH	17H	0D7FH	Adresse physique d'un module
8H	0C8FH	18H	0D8FH	Adresse physique d'un module
9H	0C9FH	19H	0D9FH	Adresse physique d'un module
AH	0CAFH	1AH	0DAFH	Adresse physique d'un module
BH	0CBFH	1BH	0DBFH	Adresse physique d'un module
CH	0CCFH	1CH	0DCFH	Adresse physique d'un module
DH	0CDFH	1DH	0DDFH	Adresse physique d'un module
EH	0CEFH	1EH	0DEFH	Adresse physique d'un module
FH	0CFFH	1FH	0DFFH	Adresse physique d'un module

Remarques :

- A la mise sous tension ou après un BREAK, toutes les adresses de déclaration sont mises à zéro.

- Si une adresse physique d'un module n'est pas validée (bit 7 de l'adresse physique), elle n'est pas prise en compte et toutes les déclarations suivantes d'adresses physiques sont ignorées (même si leur bit 7 est validé).

L'instruction de déclaration est : OUT,CH,AdDé,AdPh,....

AdDé = Adresse de déclaration d'un canal logique (mémoire de la carte MRUC 20)
AdHy = Adresse physique (bit de validation, N° de buffer et N° d'esclave du module)

Exemple de déclaration :

Affecter respectivement aux adresses logiques 0,1 et 2 (sur CAN 0), les buffers 0 des blocs MRES 22 4,5 et 0.

Mnémonique : OUT,CH,FH,84H,1FH,85H,2FH,80H.
Trame : INS,22H,CH,FH,84H,1FH,85H,2FH,80H.

Soit : <0C0FH> = 84H (bit 7=1 et code binaire du module 4)
<0C1FH> = 85H (bit 7=1 et code binaire du module 5)
<0C2FH> = 80H (bit 7=1 et code binaire du module 0)

4.3.2. Registres d'Entrées / Sorties

Les données échangées entre l'Unité Centrale (MRUC 20) et les modules MRES 22 sont rangées dans les registres organisés en 16 lignes de 16 octets : RC0 pour CAN 0 et RC1 pour CAN 1.

- **RC0 (CAN 0)** : adresses 0600H à 06FFH.
  <0600H à 060FH> = paramètres de la voie logique 0
  <0610H à 061FH> = paramètres de la voie logique 1
  ...
  <06F0H à 06FFH> = paramètres de la voie logique FH
- **RC1 (CAN 1)** : adresses 0700H à 07FFH.
  <0700H à 070FH> = paramètres de la voie logique 0
  <0710H à 071FH> = paramètres de la voie logique 1
  ...
  <07F0H à 07FFH> = paramètres de la voie logique FH

- **Détails des paramètres :**
  0xx0 : Status du module MRER 21 (si il est présent sur le réseau)
  0xx1 : Pointeur sur le buffer de réception du module (voie UART)
  0xx2 : Réservé
  0xx3 : Status du module MRES 22 (si il est présent sur le réseau)
  0xx4 : octet « mode de fonctionnement » du poids fort (PF) du port de sorties
  0xx5 : octet de commande du poids fort (PF) du port de sorties
  0xx6 : octet « mode de fonctionnement » du poids faible (Pf) du port de sorties
  0xx7 : octet de commande du poids faible (Pf) du port de sorties
  0xx8 : Recopie de l'état physique des sorties (Poids Fort)
  0xx9 : Recopie de l'état physique des sorties (Poids faible)
  0xxA : Recopie de l'état physique des entrées (Poids Fort)
  0xxB : Recopie de l'état physique des entrées (Poids faible)
  0xxC : Réservé
  0xxD : Réservé
  0xxE : Réservé
  0xxF : Réservé

  La commande d'une sortie est conditionnée par l'état des 2 bits :
  bit « mode de fonctionnement » et bit « commande ».

Mode	Commande	Etat de la sortie
0	0	0
0	1	1
1	0	Non modifiée (utilisée par une autre UC)
1	1	clignotante

- **Remarques :**
  - RC0 et RC1 sont remis à zéro à la mise sous tension ou après un BREAK.
  - Les registres sont manipulés par les instructions « JCB » et « SET ». Les instructions associées PTN (JCB) et PMN (SET) auront toujours le bit 7 = 1; le bit 4 définissant la voie CAN utilisée : Bit 4 = 0 => CAN 0; Bit 4 = 1 => CAN 1. (Cf manuel d'instructions).

- **Exemple d'utilisation des registres :**

  Faire clignoter la sortie 8 du bloc de sorties de la voie logique 0 sur CAN 0 (bit 0 de 604H et de 605H).
  Pour ce faire on utilise l'instruction SET (INS,1CH):
  INS,1CH,4,88H,5,88H.

  Le même résultat peut être obtenu avec l'instruction OUT (INS,22H) :
  INS,22H,6,4,1,5,1.

  Remarques :

  - Les instruction IN et OUT permettent de manipuler des données de 0000H et 2FFFH. Dans ce cas IA = 00H à 2FH (Cf. manuel d'instruction).

  Si IA = 8 , les données traitées sont entre 500H et 5FFH.
  Si IA = 5, les données traitées sont entre 800H et 8FFH.

  - Les entrées/sorties sont stables durant l'exécution du programme automate.

4.3.3. Défaut fugitif d'un module MRES 22:

Un défaut fugitif sur le module MRES 22 de la voie logique 1 se traduit par la valeur 11H dans le registre RD0 (0510H) pour CAN 0 ou dans le registre RD1 (0511H) pour CAN 1.

RD0 et RD1 sont des registres de capture de défauts.

Par ailleurs aux adresses 603H à 6F3H pour CAN 0 et 703H à 7F3H pour CAN 1 se trouve les status des différents blocs MRES 22 défini sur le réseau.

ANNEXES

CONTRAINTES PHYSIQUES DU RESEAU CAN

Câbles préconisés

Les informations délivrées dans cette annexe sont fonction du câble préconisé.

Identification du câble:

Marque : BELDEN
Réf. : 3084A

Affectation des fils :

Paire Rouge/Noir = paire de puissance
Rouge = + 24 Volt
Noir = 0 Volt
Paire Blanc/Bleu = paire de signal
Blanc = CAN H
Bleu = CAN L

Topologie d'un réseau CAN

Le réseau CAN est constitué d'un bus linéaire et de ramifications.

- Le bus principal doit comporter 2 résistances de terminaison de 120Ω à chaque extrémités; la longueur maximum autorisée entre les 2 noeuds les plus éloignés du réseau étant de 100m.
- Les ramifications ont une longueur autorisée de 6 mètres maximum sans dépasser 39 mètres de longueurs cumulées.

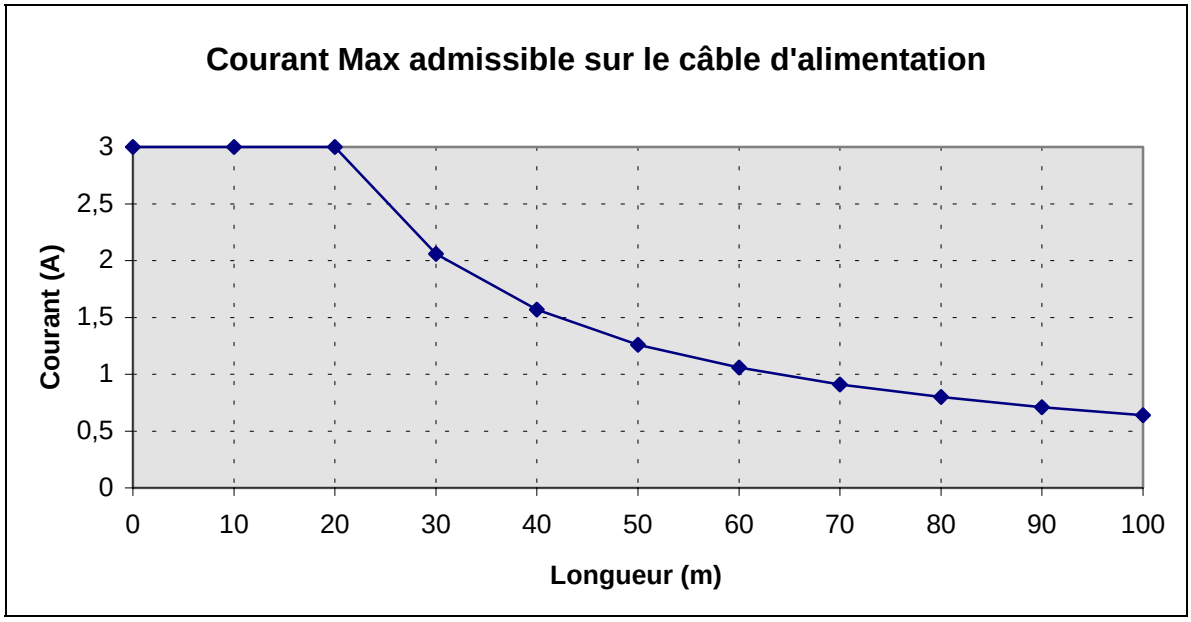
Caractérisation des courants consommés sur le réseau

Il est conseillé de vérifier la puissance de l'alimentation 24Vcc et les courants max. admissibles par le câble (fonction de la longueur du bus principal) car l'alimentation des modules et des E/R est transportée par le réseau.

Pour définir la consommation sur le réseau :

- Consulter les fiches techniques MRUC20, MRER21, MRES21 et des E/R utilisés.
- Additionner les consommations de chacun des éléments présent sur le réseau et vérifier à l'aide du tableau ci-après que les valeurs limites ne sont pas dépassées.

Longueur (m)	0	10	20	30	40	50	60	70	80	90	100
Courant (A)	3	3	3	2.06	1.57	1.26	1.06	0.91	0.80	0.71	0.64


Courant Max admissible sur le câble d'alimentation
Courant (A)
3
2,5
2
1,5
1
0,5
0
0
10
20
30
40
50
60
70
80
90
100
Longueur (m)

LE MRUC20 SOUS PROTOCOLE JBUS ® /MODBUS®

RAPPEL DU PROTOCOLE JBUS® / MODBUS ®

La taille maximale d'une trame JBUS® est de 256 octets (octets de service inclus)

Seules les fonctions de base sont implémentées:

- fonction 3 ou 4 : lecture de N mots
- fonction 6 : écriture d'un mot
- fonction 16 : écriture de N mots

FONCTION LECTURE DE N MOTS

LECTURE DE MOTS

DEMANDE :

NE	CF	Ad	.N	CRC16
1octet	1octet	2octets PF Pf	2octets PF Pf	2octets

NE : numéro du MRUC 20 sous protocole JBUS
CF : code de fonction 3 ou 4
Ad : adresse du premier mot à lire
N : nombre de mots à lire < 124

REPONSE :

NE	CF	BC	D0.....	Dn	CRC16
1octet	1octet	1octet	2octets....	2octets	2octets

BC : nombre d'octets lus (1 octet)
D0 : valeur du premier mot lu (PF = 1 er octet ; Pf = 2 éme octet ...)
Dn : valeur du dernier mot lu

JBUS ® : marque déposée de APRIL
MODBUS ® : " " de GOULD MODICON

FONCTION ECRITURE DE N MOTS

DEMANDE :

NE	CF	Ad	N	BC	D0.....	Dn	CRC16
1 octet	1 octet	2 octets PF Pf	2 octets PF Pf	1 octet	2 octets	2 octets	2 octets

NE : numéro du MRUC 20 sous protocole JBUS ®
CF : code de fonction 16(10H)
Ad : adresse du premier mot à écrire
N : nombre de mots à écrire (N < 120)
BC : nombre d'octets (BC = 2N : BC<240)
D0 : première valeur à écrire
Dn : dernière valeur à écrire

REPONSE :

NE	CF	Ad	N	CRC16
1octet	1octet	2octets	2octets	2octets

Ad : adresse du premier mot écrit
N : nombre de mots écrits (N<120)

FONCTION ECRITURE DE 1 MOT

DEMANDE

NE	CF	Ad	D	CRC16
1octet	1octet	2octets PF Pf	2octets PF Pf	2octets

NE : Numéro d'esclave
CF : code de fonction = 6
Ad : adresse du mot à écrire
D : donnée (mot) à écrire

REPONSE :

NE	CF	Ad	D	CRC16
1octet	1octet	2octets	2octets	2octets

Quand un défaut est constaté, le MRUC 20 substitue à sa réponse normale le message de défaut suivant :

MESSAGE D'ERREUR :

NE	CD	CE	CRC16
1octet	1octet	1octet	2octets

NE : numéro d'esclave
CD : code de fonction demandée augmenté de 128
CE : code erreur :

- 1 : fonction inconnue
- 2 : adresse incorrecte ou défaut du format
- 3 : donnée incorrecte
- 4 : automate non prêt (ou étiquette non présente).
- 8 : défaut d'execution de la requête

Nota : En cas de CRC16 incorrect, le MRUC 20 ne répond pas à la question posée